Pet Lab

Today we will explore OOP inheritance and how it can be used properly in class hierarchies. Generally, we can use inheritance when we can describe the parent class and child class using the **IS-A** relationship. Complete the following:

1. View the code for the Pet class into a class. Update Main to test the different methods of Pet
    - Is it fair to say that a dog **IS-A** pet? For our purposes, let's say yes; so Dog **extends** Pet. A Dog will then also have a name, a weight, and a happiness level, but those are defined in the super class and should be sent to the Pet constructor. Also add a field private String breed, that is specific to the dog class.
2. Write a constructor for Dog that calls the super class constructor with name and weight, and assigns the breed field.
3. Write at least two methods for the Dog class that are unique to dogs, but not all pets.
    - For example, dogs can go on a walk, but fish usually cannot. Note: If you change the value of happiness or weight you must first getWeight() or getHappiness(), then update the value before using setWeight() or setHappiness() methods of Pet. For example:

```
public void walk(int t)
{
    double h = getHappiness();
    double w = getWeight();

    // your calculations here

    setHappiness(d);
    setWeight(w);
}
```

4. Instantiate a Dog object in your PetTest class and test all available methods
5. Add either a Cat class that extends Pet, or a GuardDog class that extends Dog. Each class would have its own methods, but also be able to use any methods from its super class!